Developers Guidelines

October 2004

UIQ C++ and PersonalJava

P800, P900 and P910 series

Sony Ericsson

Preface

Purpose of this document

This document describes the **UIQ C++ and Personal Java** support for the Sony Ericsson P800, P900 and P910 series. Readers benefiting from this document include:

- Business decision-makers
- Corporate buyers
- IT professionals
- Support engineers
- Software developers

For more up-to-date information about technologies, products and tools refer to Sony Ericsson Developer World.

These Developers Guidelines are published by:

Sony Ericsson Mobile Communications AB,
SE-221 88 Lund, Sweden

Phone: +46 46 19 40 00
Fax: +46 46 19 41 00
www.SonyEricsson.com/

Third edition (October 2004)
Publication number: EN/LZT 108 6846, R3A

This document is published by Sony Ericsson Mobile Communications AB, without any warranty*. Improvements and changes to this text necessitated by typographical errors, inaccuracies of current information or improvements to programs and/or equipment, may be made by Sony Ericsson Mobile Communications AB at any time and without notice. Such changes will, however, be incorporated into new editions of this document. Printed versions are to be regarded as temporary reference copies only.

*All implied warranties, including without limitation the implied warranties of merchantability or fitness for a particular purpose, are excluded. In no event shall Sony Ericsson or its licensors be liable for incidental or consequential damages of any nature, including but not limited to lost profits or commercial loss, arising out of the use of the information in this document.

Document history

Change history		
2003-10-31	Version R1A	First release
2004-08-18	Version R2A	Second release, P910 updates
2004-10-11	Version R3A	Third release, updates and editorial changes

Trademarks and acknowledgements

Sun, Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Symbian and all Symbian-based marks and logos are trademarks of Symbian Limited.

Borland and JBuilder are trademarks or registered trademarks of Borland Software Corporation in the United States and other countries.

Metrowerks and CodeWarrior are trademarks or registered trademarks of Metrowerks.

The Bluetooth word mark and logos are owned by the Bluetooth SIG, Inc. and any use of such marks by Sony Ericsson is under license.

Microsoft, Windows and Visual Basic are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.

Other product and company names mentioned herein may be the trademarks of their respective owners.

Terminology and abbreviations

API
Application Programming Interface

CLDC
Connected Limited Device Configuration. A J2ME configuration for mobile devices.

HTTP
HyperText Transfer Protocol

J2ME
Java 2 Micro Edition. A Java platform targeting ‘micro’ devices with small processors and memory capacities, such as mobile phones, communicators and PDAs.

JSR
Java Specification Request

MIDlet Suite
One ore more MIDlets that belong to the same JAR file and share common date

MIDP
Mobile Information Device Profile. A J2ME profile connected to the CLDC for mobile phones

MMAPI
Mobile Media Application Programming Interface

SMS
Short Message Service

URI
Uniform Resource Identifier

URL
Uniform Resource Locator

WMA
Wireless Messaging API

WTK
Wireless Toolkit

Online Developer Resources

On the Sony Ericsson Developer World website, http://www.sonyericsson.com/developer/, developers will find all documentation and tools such as phone White Papers, Developers Guidelines, SDK's and API's, and so on. The developer web site also contains discussion forums monitored by our Sony Ericsson Developer Support team, a searcheable Knowledge Base of support queries and solutions, Tips & Tricks, example code, and so on. To stay up-to-date on development issues, register and subscribe to the monthly Sony Ericsson Developer Newsletter.

Sony Ericsson Developer Support

Sony Ericsson offers developers professional technical support services. The service can be purchased from the developer web portal, as part of the Sony Ericsson Core and Core+ membership package, or as individual support incidents. There are two levels of support included in the memberships.

The **Basic E-mail Developer Support** is an annual support service included in the Core membership that provides developers with all the basics to successfully develop world class applications for Sony Ericsson products. With this support contract, developers get access to Sony Ericsson developer support engineers via email with same-day response, five technical support incidents as well as the ability to purchase more.

The **Priority E-mail Developer Support** is an annual support service included in the Core+ membership that equips professional developers with everything they need to successfully develop world-class applications for Sony Ericsson products. With this support contract, developers get priority access to Sony Ericsson developer support engineers via email with fast response times and up to 50 technical support incidents.

Contents

Technical overview and architecture

This document is valid for the Sony Ericsson P800, P900 and P910 series mobile phones.

The Sony Ericsson P800, P900 and P910 series mobile phones are based on the Symbian™ UIQ user interface. UIQ is a pen-based user interface platform for media rich mobile phones based on the Symbian Operating System (OS).

The P800, P900 and P910 series is a versatile application platform enabling application developers to create applications in a variety of programming languages, including native C++, Personal Java (pJava), Java (J2ME) and Visual Basic. This document provides guidelines for developing C++ and pJava based applications. For J2ME and Visual Basic development environments, refer to the links provided at the end of this document.

The reader of this document is expected to understand the basics of Symbian OS™, UIQ and the development environment of UIQ SDK. For further reading, please refer to http://www.uiq.com/developer.

A developer's perspective on the P800, P900 and P910 series

The P800, P900 and the P910 series are based on Symbian OS 7.0. The P800 User Interface (UI) is based on UIQ 2.0, whereas the P900 and P910 series are based on UIQ 2.1. The table below presents a comparison of what is supported by the different phones and what developers need to consider when designing applications:

Technical data	**P910 series**	**P900 series**	**P800 series**
SDK support	UIQ 2.1, 2.0	UIQ 2.1, 2.0	UIQ 2.0
Colour screen	262K	65K	4K
Camera API	Yes	Yes	Yes (CXC 162001 R2A14 and later)
Camera Video API	Yes	Yes	No
MMS API	Yes	No	No
Vibration API	Yes	Yes	Yes
Streaming AMR API	Yes	Yes	Yes (CXC 162001 R2A14 and later)
Themes	Yes	Yes	No
Internal User storage	64 MB	16 MB	12 MB
Additional storage: Memory Stick Duo Memory Stick Duo Pro	 supports up to 128 MB supports up to 1 GB	 supports up to 128 MB No	 supports up to 128 MB No

Application development

This chapter contains information for the developer of C++ and PersonalJava™ applications for the Sony Ericsson P800, P900 and P910 series of mobile phones.

General information about the P900 series

The table below lists the major features of the P900 series:

Feature	Support
Third party application support	• C++ • PersonalJava • J2ME^{TM} CLDC/MIDP 2.0 with JSR 120 and JSR 82. • Microsoft® Visual Basic® 6
Music Player	MP3,WAV, AU, AMR, MIDI (G-MIDI level 1 with 24 voices polyphony), RMF, iMelody
Video Recorder and Player	3GPP and MPEG4 (VGA, QVGA, QQVGA and 24-bit colour depth) with AMR encoded audio (2 quality levels + video without audio)
Streaming	RTSP according to 3GPP
Still images	JPEG/EXIF
Feature	**Support**
Integrated browser	• WAP Version:2.0 • Markup HTML3.2 (excl. JavaScript and partial support for Frames) • Languages: WML 1.3 WBXML XHTML Basic XHTML Mobile Profile cHTML • Scripting: Compiled WML scripts • Style sheets: WCSS • Security: WTLS Class 1, 2, 3 TLS/SSL • Certificates: Pre-install & download WTLS, X.509 • WIM: WIM interface including SIM-WIM

Note: For further J2ME related information, see “Developers Guidelines Java MIDP 2.0 for the P900 series mobile phones” available at Sony Ericsson Developer World.

General tips for the P900 and P910 series

The UIQ version 2.1 introduces new features within the Symbian OS platform:

- Make sure your AppUi is derived from CQikAppUi (Qikon) and not from CEikAppUi (Uikon).
- Ensure that your application is “theme aware”. Use logical colors to draw application controls and propagate color scheme change messages through your application.
- Add a large version of your application icon for display in the Applauncher icon view.
- The media server supports MP3 audio files. If your application uses audio files in any format, other than AMR, then convert them to MP3.

Internal user storage – C: drive

The user storage space (C: drive) is shared across applications without any imposed restrictions, apart from the space becoming full. Unlike a PC, the user does not need to be aware of the underlying file system. Applications will always make sensible choices and store information automatically, simplifying management of data. Third party applications may implement more complex file management solutions where required.

Applications and information are placed in the internal storage of the P800, P900 and P910 series in the factory. This provides sample demonstration, educational, multimedia and fun content so that the P800/ P900 series can be used directly out of the box. Much of this can be deleted by the user in order to make the space available for personal use. The following may be deleted by the user:

- Sony Ericsson Multimedia Content
  (Sound, pictures, video and other files chosen to show off the capabilities of the P800, P900 and P910 series)
- Demonstration application files are available for the P800, P900 and P910 series. These files may be uninstalled.
- In the **P800 series**, Chess and Solitaire may be uninstalled if more space is needed.
  In the **P900 and P910 series** these games **do not** utilise internal user storage.
- Language files (Language is selected at First Startup and Master Reset).
  If the user selects any language other than UK English, the selected language files and UK English are retained on the C: drive and the other language files are deleted.)

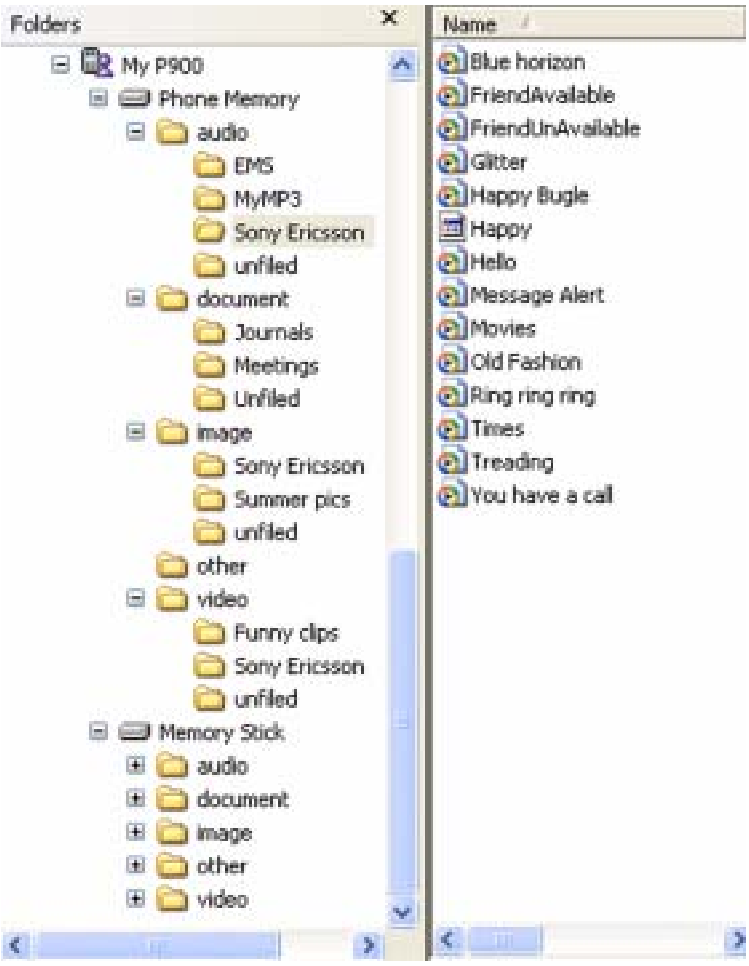
Note: Themes can **not** be deleted.

Folder view of internal storage

This section explains in more detail how the data is organised on the C: drive.

Beneath each media type and documents, the material on the mobile phone is broken into Folders. For each named folder in the related application, a folder is created beneath the applicable media type. For example, if the user creates a ‘My MP3’ folder in the Music player application, a ‘My MP3’ subfolder is created beneath the audio folder. **Unfiled** folders are created in the initial folder structure and all material is placed in the **unfiled** folder by default. Sony Ericsson Multimedia Content is stored in ‘Sony Ericsson’ sub-folders.

When the file system is viewed from a PC using PC Suite, the C: drive is called 'Phone Memory'. Only a subset of the folders is accessible. The picture below shows the layout of the folder structure:



Folders

- My P900
  - Phone Memory
    - audio
      - EMS
      - MyMP3
      - Sony Ericsson
      - unfiled
    - document
      - Journals
      - Meetings
      - Unfiled
    - image
      - Sony Ericsson
      - Summer pics
      - unfiled
    - other
    - video
      - Funny clips
      - Sony Ericsson
      - unfiled
  - Memory Stick
    - audio
    - document
    - image
    - other
    - video

Name

- Blue horizon
- FriendAvailable
- FriendUnAvailable
- Glitter
- Happy Bugle
- Happy
- Hello
- Message Alert
- Movies
- Old Fashion
- Ring ring ring
- Times
- Treading
- You have a call

- Multimedia is stored in a structure where each media type has its own folder: ‘audio’, ‘video’ and ‘image’.
- Documents (such as Microsoft Word files) are stored under the ‘document’ folder.
- An ‘other’ folder provides a place for files that do not fit into the other categories. Beneath each media type and documents, the material is broken into folders.

C++

Applications written in C++ have the most direct access to hardware and functions of the phone and require in-depth knowledge of Symbian C++. Since C++ is a compiled language, the applications run faster and have a faster startup time than those written in pJava, J2ME or Visual Basic. This does not imply that applications written in other languages will be slow. In fact, the other languages may be a better choice if your application does not require excessive speed performance.

Data calls can be made with C++ applications. Low level access to the media player, screen, Bluetooth™, and so on, is available.You can also utilise the full screen and your applications will not be restricted to the canvas window of pJava and J2ME.

Sony Ericsson C++ API extensions for the UIQ SDK are available for download from Sony Ericsson Developer World. These APIs enable developers to write UIQ applications to access:

- camera/video
- streaming AMR(Adaptive Multi-Rate speech codec)
- vibration functions

The P800, P900 and P910 series mobile phones support several productivity enhancing tools - Integrated Development Environment (IDE). Below is a list of supported IDEs for developing both emulated and native C++ applications.

IDE	SDK compatibility	To be used with...
CodeWarrior™ Development Studio for Symbian OS 2.5	• UIQ 2.0 SDK • Sony Ericsson API extensions (camera/video,vibration and AMR streaming)	P800 series C++ native application development
CodeWarrior Development Studio for Symbian OS 2.5	• UIQ 2.1 SDK • Sony Ericsson API extensions	P900 and P910 series C++ native application development
Borland® C++ BuilderX Mobile edition	• UIQ 2.1 SDK • Sony Ericsson API extensions	P900 and P910 C++ native application development

Note**:** An IDE is required if you need to test your application in the UIQ emulator. Without these productivity tools you will only be able to test your application on the target device itself.

PersonalJava

This section offers advice about developing PersonalJava applications. This requires Symbian OS environment specific knowledge not needed for MIDP application development.

PersonalJava (referred to as “pJava” in this document) was one of the first Java programming environments targeted for development of applications for resource-constrained devices. These devices include PDAs (Personal Digital Assistants) and Communicators/Smartphones. pJava specifies a reduced set of class libraries compared to a desktop environment and historically represented the first attempt at selecting desktop level APIs suitable for the Java runtime environment of a high-end mobile device.

Over the years the Symbian port of pJava to Symbian OS has been undergoing substantial optimization (assembler coded byte code interpreter, optimized AWT library and so on) that today, in combination with hardware performance advancements, make this a powerful alternative for realizing mobile applications. pJava is supported on both P800, P900 and P910 series, and is provided as an alternative to using Symbian OS C++ for feature-rich third party applications.

Applications developed in pJava do not have access to the hardware and functions of the phone unless you use the Java Native Interface (JNI) and write the access procedures in C++. Your applications written in pJava will probably run slower than a corresponding application written in C++, however they are typically easier to write. As previously stated, this does not mean to imply that pJava applications will not perform well. You can access the features of the phone by using JNI to access C++ DLLs to access the features of the phone available to C++ applications.

Note:
JavaPhone APIs is not available on P800, P900 and P910 series and there is no direct Bluetooth access.

For more information on programming in pJava, download the self training course from Sony Ericsson Developer World, see links.

Additional tools useful to the pJava developer

- The Sun Microsystems Inc. specification of Personal Java (PJAE). This document describes the subset of a JDK 1.1.x generation class library that is supported by a pJava edition. The P800, P900 and P910 series pJava runtime environment corresponds to version 1.1.1a (http://java.sun.com/products/personaljava/spec-1-1-1/pjae-spec.pdf)
- A copy of the Sun Microsystems Inc. “JDK” (Java Development Kit).
- If you intend to use the JDK standalone to carry out the compilation of Java source code from a DOS command shell to be executed on P800, P900 and P910 series, you should download and install the JDK.1.8_010 at http://java.sun.com/products/archive/jdk/1.1.8_010/index.html, which closely corresponds to the pJava evolution. It has formally passed the “Sun End of Life process” but may still be preferred for this usage, since it does not offer more advanced Java features and class library than supported by the Symbian Java runtime, thus reducing the risk of code incompatibility.

- It is recommended to install Sun J2SDK. When using the J2SDK, it is important to compile using a pJava “classes.zip” file. Such a file can be found in the UIQ SDK. Also, the javac compiler directive “-target 1.1” must be used for the J2SDK to assure that JDK 1.1.* compatible bytecode is generated. A good example of this can be found on the Sun web page: http://java.sun.com/j2se/1.5.0/docs/tooldocs/windows/javac.html#crosscomp-example.

- JavaCheck(tm) is a useful tool provided by Sun that helps you to find out if your source code conforms to the pJava specification. See:
  http://java.sun.com/products/personaljava/javacheck.html
  You should use JavaCheck 3.0 which corresponds to the 1.1.x PJAE specification.

pJava performance

pJava is located on the A: drive in the P800 series. However, in the P900 and P910 series, pJava resides in ROM (Z: drive) and 'executes in place'. This means that in the P900 and P910 series, the virtual machine binaries and classes.zip are not copied to RAM before execution. This results in approximately 1 second shorter startup time for an application and lower RAM usage per launched application.

pJava application considerations

The available screen size is 276x208 pixels and applications are usually based on the java.awt.Frame class. A good rule to minimise the start-up of applications containing complex GUIs, is to instantiate all GUI elements just before they are first needed, rather than at start-up. To give the user a perception of a quick starting application, it may be a good idea to load and display a “splash screen” as the first action while additional initialisation continues. This is something to consider for commercial grade applications where this additional development work is well worth the effort.

pJava font limitations (java.awt.Font)

The fonts and font sizes available to pJava applications are limited by the native fonts available in Symbian OS.

It is possible to obtain a list of the available fonts by using the java.awt.Toolkit class. The toolkit is acquired by using:

```
java.awt.Tookit tk=java.awt.Toolkit.getDefaultToolkit();
```

You can now get a string array of available fonts by using:

```
String[]fonts = tk.getFontList();
```

Only a limited set of font sizes are available for each font. For pJava, not all font sizes that can be specified in the font constructor are available. Instead the nearest smaller size available will be constructed. It is possible to sense the actual size obtained by using the method:
FontMetrics fm = tk.getFontMetrics();

Behaviour related to flip modes

The P800, P900 and P910 series operates in two distinctly different UI modes: Flip Open (FO) and Flip Closed (FC). pJava applications can be launched and can only appear on screen while in the FO mode. On transition from FO to FC, the UI of any pJava application being executed becomes out of focus (hidden) and the user is unable to interact with the application. The application may however continue to execute and perform various tasks in the background. The user is able to return to the pJava application by returning to the FO mode Application Launcher panel and selecting the application from the list of installed applications. If that pJava application is already running, its UI appears instantly on screen. If it is not running, the application needs to start up before it can appear on screen.

To detect an AWT window state transition for pJava(AWT), it is possible to override the methods WindowClosing and WindowOpening in java.awt.event.WindowAdapter. If such a transition occurs, the application may decide to perform certain actions such as terminating, reducing resource usage, saving/restoring state and so on.

Application development

This section provides a high level overview of the components required to successfully develop P800, P900 and P910 series applications.

The UIQ SDK is generic and is not tailor-made for the Sony Ericsson smartphones. The P800, P900 and P900 series devices are platforms based on the Symbian/UIQ base port supporting most of the APIs available in the SDK.

UIQ 2.0 vs UIQ 2.1 and application compatibility

The table below lists the UIQ versions and the appropriate development platform:

Use UIQ version...	for development on the ...
UIQ 2.1 SDK	P900 and P910 series
UIQ 2.0 SDK	P800 series

Note:
It is recommended that you test your application on all target devices.

Convention

From this point on, the UIQ 2.0 SDK and UIQ 2.1 SDK will be referred to as UIQ SDK meaning the SDK applicable to the platform in reference.

Sony Ericsson publishes - at its own discretion - extra platform specific APIs. These are available from Sony Ericsson Developer World.

UIQ SDK for Symbian OS v7.0

The UIQ SDK contains:

- Library files
- Documentation
- Sample code
- Tools for packaging applications into Symbian installable format (.sis) for C++ and Personal Java applications.
  These .sis files can be compared to 'InstallShield' in Microsoft® Windows® development environment.
- A standard UIQ device emulator to simplify and speed up your development process
- Deployment tools, such as AIFBuilder and MakeSIS

Technical requirements for the UIQ SDK

- Windows NT 4.0 or Windows 2000. The SDK has not been thoroughly tested and verified with Windows XP.
- Java Runtime Environment (JRE) version 1.3.1 (included in the SDK) is required for installing the SDK. The installation process has been verified with this JRE.
- 1,4 GB or more of free disk space (to install all components).
- WinZip or an equivalent file utility for unzipping files.

Note:
The UIQ SDK needs to be installed before installing Metrowerks CodeWarrior Development Studio for Symbian OS.

Device Emulator and Library Linking

The UIQ SDK includes a generic UIQ emulator enabling application developers to emulate and debug applications on a PC during the development cycle.

The directory ***\epoc32\tools*** is required for the ***bldmake*** command. This command generates the ***abld*** batch file used in building the specific project.

Library linking for applications with device emulation

When developing for the Emulator, you need to link to the libraries in:

- the subdirectory **epoc32\release\WINSCW** if using Metrowerks CodeWarrior.
- the subdirectory **epoc32\release\WINS** if using Borland C++ BuilderX.

Library linking for applications for target execution

When building your application for the target device/mobile phone, you need to link to the libraries in:

- the subdirectory**epoc32\release\THUMB*.***

Skin images

Sony Ericsson provides skin images and property files for changing the “look and feel” of the generic UIQ emulator of the Sony Ericsson P800, P900 and P910 series. The skin images are available at Sony Ericsson Developer World.

In order to mirror the appearance of the phone’s screen in the UIQ Emulator, do the following:

1. Backup existing files in the UIQ SDK.

   <sdkroot>\epoc32\data\epoc.bmp
   and
   <sdkroot>\epoc32\data\epoc.ini

2. Replace the above epoc.bmp and epoc.ini files with the files provided by Sony Ericsson.

Emulator issues and guidelines

- An emulator error may occur for PCs with more than two COM ports. Removing `irda.esk` and `bt.esk` from **\epoc32\[target]\system\data** resolves this problem.

- If the emulator does not start when pressing run and the command line window says: "`Unable to initialize threads: cannot find class java/lang/Thread`", set the environment variable **_epoc_drive_j** to **<SDK_DIR>\epoc32\java\**.

- Personal Java applications can be compiled with any Personal Java compiler and tested in the UIQ emulator.

- C++ applications must be compiled with CodeWarrior Development Studio for Symbian OS or Borland C++ BuilderX Mobile Edition to be able to be tested in the UIQ emulator. See page 12 for further information concerning IDEs.

Chinese extensions to the emulator

If your applications require Chinese Traditional and Simplified Chinese input in the emulator (P802/P908), a Chinese extension to the UIQ SDK can be downloaded from Sony Ericsson Developer World.

Porting applications

The UIQ User interface design is referred to as Qikon. Another well-known User Interface in the Symbian community is Series 60, referred to as Avkon.

Porting guidelines are available for developers who would like to port their applications from Series 60 to UIQ. There are also commercial porting tools and professional services available for developers who want to port from Avkon to Qikon user interfaces. More information can be found in “Porting Guide S60 to P800/P900” at Sony Ericsson Developer World.

Sony Ericsson specific APIs

The Sony Ericsson specific API:s and relevant documents can be found at Sony Ericsson Developer World.

Camera API

The Sony Ericsson Camera APIs for the P800, P900 and P910 series support:

- Still image capturing of RGB bitmap based pictures in three sizes.
- Still image mode direct-to-screen view finding of one fixed size (QQVGA). The Camera API for the P900 also supports video recording of QCIF sized frames in YUV 4:2:0 format at 15 frames per second and video mode direct-to-screen view finding of one fixed size (QCIF).

Detailed API information is available within the downloadable packages.

The Camera API support for the P800 series is not pre-installed in the device and must be installed separately. At install time the package (ECam.sis) checks to see if the target phone supports the Camera API. Applications using the Camera API embed ECam.sis into the application package.

Compatibility

The camera API for the P800 is compatible with P800 Organiser software CXC162001 R2A14 and later.

There is no need to install any Camera API packages for the P900 and P910 series. This function is provided in the base software.

To make the Camera API a part of an installable application, add the following line to the application's .pkg file:

```
@"ECam.sis", (0x101F7D30)
```

Vibration API

The Sony Ericsson Vibration API turns the vibration function on and off. This API is also used to specify the time to turn the vibration on or off and the number of times to run this scenario. Detailed API information is available within the downloadable package.

The Vibration API in the P900, P910 and the P800 series are binary compatible. The dynamic link library is present in the P900 and P910 series but needs to be installed manually in the P800 series.

The Vibration.sis file has to be embedded into the application .sis-file to provide application support on both P800, P900 and P910 series. This embedded .sis-file will be ignored by the P900 and P910.

To make the Vibration API a part of an installable application, add the following line to the application's .pkg file:

```
@"Vibration.sis", (0x101F94A3)
```

Streaming AMR API

AMR is standard speech GSM encoding/decoding technology. It lets you play and record AMR encoded sound at different bit rates. This function can therefore be used by your application.

The Streaming AMR API support is not supplied in the base software of the P800 and must therefore be installed separately. The package provided (StreamingAmr.sis) is used to add this Streaming AMR API to the device.

Compatibility

The Streaming AMR API is compatible with the P800 Organiser software CXC162001 R2A14 and later.

There is no need to install any AMR packages for the P900 or P910 series. This function is provided in the base software.

To make the Streaming AMR API a part of an installable application, add the following line to the application's.pkg file:

@"StreamingAmr.sis", (0x101F94AE)

MMS API

The P910 incorporates its own version version of MMS APIs allowing multimedia messages to be received on and sent from the phone. This MMS is different from the MMS service that is found in the standard UIQ 2.1 SDK.

The MMS API is based on a similar MMS API available for Nokia Series 60 Developer Platform 2.0. This will allow easy porting of applications that use MMS between Series 60 and UIQ 2.1.

Compatibility

The MMS API is compatible with the P910 Organiser software CXC 162041 R2A17 and later. The MMS API will **not** work on P800 and P900 series

Other useful Symbian APIs

Device identification

HAL can be used to identify and adjust the behaviour of the P800, P900 and P910 series applications.

```
#include <hal.h>

HAL::Get(HALData::EMachineUid, aInt)
```

The aInt values are listed in the table below:

Device	Device UID
P910	`aInt = 0x10200AC6`
P900	`aInt = 0x101FB2AE`
P800	`aInt = 0x101F408B`
UIQ emulator	`aInt = 0x10005F62`

Bluetooth

For more information, see the C++/Bluetooth Special Interest Paper at Sony Ericsson Developer World.

IMEI, IMSI

The **IMEI** number can be obtained by reading ***C:\System\plpvar.ini***

The **IMSI** number can be obtained by reading ***C:\System\data\imsi.txt***

Colour display

The P910 has a colour-depth of 262K, the P900 has a colour-depth of 65K and the P800 4K. Applications requiring direct access to the screen will encounter problems with devices. It is therefore recommended that applications use the **CDirectScreenAccess** method. This method is described with an example in the SDK in:

> **<SDKROOT>\examples\Graphics\WS\Direct\**
> and
> **<SDKROOT>\examples\Graphics\2DAccel**

The latter shows how to query the system to obtain the resolution used by the device.

Audio

Programmers can use the versatile audio engine available in the P800, P900 and P910 series mobile phones. The mobile phones use Beatnik Audio Engine, Mini Edition (MiniBAE) to reproduce high quality MIDI sounds and is able to play different media types together, at the same time.

To extend your applications with high quality sound effects, use the miniBAE devkit available from Sony Ericsson Developer World.

Media server

The Media server is ‘single threaded’. This means that audio recording/playback and image operations cannot be done at the same time without affecting each other. Applications that use the Media Server should take this into consideration and ensure that they break up long operations (for example, converting a GIF to a BMP) into smaller steps. In this way, the Media Server is able to serve both the audio and the image requests in a reasonable amount of time.

API Limitations

Some components in the P800, P900 and P910 series have been replaced and therefore some of the standard Symbian/UIQ APIs are not valid.

COMMDB Object

The following interface has a slightly different behaviour:

CCommsDbTableView::ReadTextL(const TDesC& aColumn, TDes8& aValue)

The difference relates to the type of data stored in 'aColumn'.

In the SDK version of COMMDB, 'aColumn' is either narrow text (EDbColText8) or wide text (EDbColText16). If it is wide text, a conversion is performed.

In the P800, P900 and P910 series version of COMMDB, 'aColumn' must be narrow. If it is not, the client will receive the following error:

Category - DBMS
Reason - 7 (EDbWrongType)

BIOMSG Object

The following MTM library is **not** present in the P800, P900 and P910 series:

Z:\SYSTEM\PARSERS\wapp.dll

Any client attempting to dynamically load this library will receive the error code -1 (KErrNotFound).

BAFL bassnd.h

Built in system sounds, i.e. in ROM, cannot be replaced by using the function:

BaSystemSound::SetSoundL()

CEikInfoDialog

Subclassing of CEikInfoDialog is not recommended in the P800, P900 and P910 series.

WAP stack

The component does not exist in the P800, P900 and P910 series – do not use these APIs. Another unpublished set of APIs are used.

ETel

ETel Core can be used to produce a simple telephony application. It handles placing of a call, answering a call and ending a call. It does not use GSM specific information such as signal strength and network name. The functions of the ETel Core API which are supported are listed below:

Class **RCall** (functionality associated with a specific call)	Class **RLine** (functionality associated with a specific line)	Class **RPhone** (functionality associated with a specific phone)
AnswerIncomingCall Connect Dial GetCallDuration GetCallParams GetCaps GetFaxSettings GetInfo GetOwnershipStatus GetStatus HangUp NotifyCapsChange NotifyHookChange NotifyStatusChange GetBearerServiceInfo AcquireOwnership LoanDataPort NotifyCallDurationChange RecoverDataPort ReferenceCount SetFaxSettings TransferOwnership	EnumerateCall GetCallInfo GetCaps GetHookStatus GetInfo GetStatus NotifyCallAdded NotifyCapsChange NotifyHookChange NotifyIncomingCall NotifyStatusChange	GetLineInfo GetStatus EnumerateLines GetCaps GetInfo Initialise NotifyCapsChange NotifyModemDetected

Messaging/MMS

The P800 and P900 series do not use the standard Symbian MMS API.

Deploying applications on the P800, P900 and P910 series

The steps described below are documented in the UIQ SDK. It is important that the developer is familiar with the “Aifbuilder” and “MakeSIS” tools which are used for preparing applications to execute on the mobile phones.

Each Symbian OS UI variant has a shell program that allows the user to run application programs. For an application to be run from such a shell program, at least two files are needed:

1. An application file (**.app**) minimally containing a unique identifier (UID) to identify the program.

2. An application information file (**.aif**) specifying the application's icon and caption in the P800, P900 and P910 series Application Launcher list of applications.

Note:
There is no need to generate an .aif file. If this file does not exist, the application uses the default system icon. It is recommended that each application has a resource file and an .aif file.

Other files may also be required depending on the programming language. For a pJava application, an additional text file (.txt) is required to specify the command line to the JVM when the application is launched.

- Create a .pkg file by copying a sample one from the Install directory <SDK_DIR>\Examples\ToolsAndUtilities\Install\ and modify it to match your application.
- Run the MakeSIS tool from the command line.

"MakeSIS" is needed for generating .sis format installation files to ensure availability of a C++ and pJava application to end users for installation.

P800/P900 series specific .sis packages

The P800, P900 and P910 series are open devices for adding applications. Today, there are Symbian devices on the market based on different User Interfaces (UI), namely UIQ, Series 60, Series 80 and Series 90. Applications for these devices are all installed using an installation package.

The format is developed by Symbian and is called **.sis**.

To ensure that the appropriate **.sis** file is installed on a designated target device, the P800, P900 and P910 series **.sis** packages need to contain platform-specific package information. If this information is not included, the installation will stop and the user will receive the error [Incompatible installer version] when trying to install the .sis package.

The table below describes the information that needs to be added to the .sis package files depending on the platform being used:

Platform	Information to be added to .pkg file	For general UIQ v2.0 compatibility
P910 series	*(0x101FBB35), 2, 0, 0, {"SonyEricssonP9xxPlatformProductID"}*	*(0x101F61CE), 2, 1, 0, {"UIQ21ProductID"}*
P900 series	*(0x101FBB35), 1, 0, 0, {"SonyEricssonP90xPlatformProductID"}*	*(0x101F61CE), 2, 1, 0, {"UIQ21ProductID"}*
P800 series	*(0x101F80BE), 1, 0, 0, {"SonyEricssonP80xPlatformProductID"}*	*(0x101F617B), 2, 0, 0, {"UIQ20ProductID"}*

Note:
Applications that are P800 series specific will run on the P900 and P910 series platform, but not vice versa.

Installation and data storage

Applications can be easily downloaded directly to the mobile phone using the browser, or may be installed from a connected PC, using USB, Infrared or Bluetooth. Applications can also be installed from a Memory Stick.

Multimedia content such as images, movie clips, sound clips and general-purpose files such as Word documents can either be stored locally (C: drive) or on a Memory Stick (D: drive). Third party (pJava and C++) applications can also use the Memory Stick for both application and data storage. These are all accessed using the viewers provided with the mobile phones.

The built in browser can download .jar/.jad or .sis files from WML and xHTML pages provided that the server supports the actual file types. The P800, P900 and P910 series mobile phones can also download .jar/.jad/.sis files from ordinary Web servers provided that they support the MIME type for each of the file types. Some servers need to be configured for this.

On-target debugging

Testing and debugging are two of the most time-intensive phases of the application development cycle. The Sony Ericsson P800 series mobile phone was the first Symbian device on the market to support on-target debugging of native C++ applications.

It is recommended that you use an IDE if you need to perform on-target debugging of your application. It is possible to do this using:

- Metrowerks CodeWarrior for Symbian OS v2, Professional and Enterprise Editions using the proprietary MetroTRK on-target debugger client in the P800/P900 series. This allows developers to debug their code on the commercial releases of the P800/P900 series. Metrowerks MetroTRK (Metrowerks target resident kernel) provides target side application level debug support for the P800, P900 and P910 series. This can be used in conjunction with the Metrowerks CodeWarrior IDE for Symbian OS. The MetroTRK description, updates and patches are available from http://www.metrowerks.com/MW/develop/wireless/default.htm.

- Borland C++BuilderX Mobile Edition. Borland Mobile studio allows on-target debugging via gdbstub. For more information, see http://www.borland.com/mobile/cbuilderx.

More information

For information on-target debugging for the P800, P900 and P910 series, refer to Sony Ericsson Developer World.

Signing digital applications

Symbian Signed

“Symbian Signed” is a new program from Symbian that supports **tested and digitally signed applications** on Symbian OS phones. It is supported by all Symbian licensees including Sony Ericsson.

A “signed” application means that the application has passed the test criteria and is digitally signed to **guarantee the source and integrity of the application**. Sony Ericsson encourages all users to use signed applications only.

Developers are allowed to use a Symbian logo called “For Symbian OS” to indicate compliance to the program. For further information see the Symbian Signed white paper.

Installing ”Symbian Signed” applications on the P800, P900 and P910 series

In order to install an application that is “Symbian Signed”, a Symbian root certificate must be present in your phone. This certificate is pre installed in the P900 and P910 series mobile phones but needs to be installed in the P800 series mobile phones by following the installation steps in the section below: “How to install the Symbian certificate**”.**

How to check if the Symbian certificate is installed

- Go to **Applications -> Control Panel -> Certificate manager -> Certificate Authority**. The list contains a certificate called “Symbian”.
- If the Symbian certificate is not there you will need to install it.
- You must have the “Sony Ericsson” certificate in the list. If you do not see the “Sony Ericsson” certificate you must contact Sony Ericsson customer support.

All future Symbian OS based phones will have a preinstalled Symbian root certificate from the factory. If you have accidentally erased it from your phone it is possible to use the same procedure as for the P800 series.

How to install the Symbian certificate

1. Download the Certificate Installation Application (CIA) to your PC. The CIA is available from the following sites:

   - Sony Ericsson’s support web site at http://www.sonyericsson.com/support
   - Sony Ericsson Developer World at http://www.sonyericsson.com/developer
   - Sony Ericsson Application Shop at www.sonyericsson.com/applicationshop
   - Symbian Signed web site at http://www.symbiansigned.com

2. Install the file **Certificate_installation.sis** to your phone through the PC suite in the same way as for all downloadable applications (the format of the application is a normal Symbian OS installation file).

3. After successful installation, the application called "Certificate installation" appears in the application list. (Scroll down if necessary).

4. Select "Certificate installation" to install the certificate. Upon successful installation, the Symbian certificate appears in the **Certificate Authority** list.

It is now possible to install “Symbian Signed“ applications in a secure way on your phone!

Go-to-Market opportunities

Sony Ericsson provides you with the opportunity to have your applications marketed through the Sony Ericsson application shop at www.sonyericsson.com/applicationshop. P800, P900 and P910 series mobile phone users are able to buy business and leisure applications which give extra functionality to their handsets.

To apply for this please refer to Sony Ericsson Developer World.

Technical specifications and references

This chapter provides technical references for the mobile phones in the scope of the document, and also information on reference documents and useful links.

Technical specifications for the P800, P900 and P910 series

Specification table:

Characteristic	**P910 series**	**P900 series**	**P800 series**
Pixel ratio (H: W)	1:1	1:1	1:1
Colours	18-bit (262 144 colours)	16-bit (65 536 colours)	12-bit (4096 colours)
Internal user storage	64 MB	16 MB	12 MB
RAM	32 MB	16 MB	16 MB
Other API support	pJava supports JNI (Java Native Interface)	pJava supports JNI (Java Native Interface)	pJava supports JNI (Java Native Interface)
pJava version	1.1.1a	1.1.1a	1.1.1a
Physical communication interfaces	Bluetooth, IR, Serial	Bluetooth, IR, Serial	Bluetooth, IR, Serial
Characteristic	**P910**	**P900 series**	**P800 series**
Bluetooth APIs	C++ and J2ME *	C++ and J2ME *	C++ only
JavaPhone API	No	No	No
Java Serial APIs	No	No	No
Debug interface	C++ only	C++ only	C++ only
OTA RP v1.0	Yes, applicable to J2ME MIDP applications	Yes, applicable to J2ME MIDP applications	Yes, applicable to J2ME MIDP applications
4 way arrow keys with select	KVM - soft buttons	KVM - soft buttons	KVM - soft buttons
Other function keys	Jog Dial (up, down, towards, away, select	Jog Dial (up, down, towards, away, select)	Jog Dial (up, down, select)
Numeric keys	KVM soft buttons 0-9, *# (ITU-T)	KVM soft buttons 0-9, *# (ITU-T)	KVM soft buttons 0-9, *# (ITU-T)
Inside flip keyboard	Yes	No	No

* For more information about J2ME functions for the P900 series, refer to “Developers Guidelines for J2ME MIDP 2.0 for the P900 and P910 series” at Sony Ericsson Developer World.

Links and references

pJava on P800, P900 and P910 series

pJava Application Environment	http://java.sun.com/products/personaljava/ Refer also to the UIQ SDK documentation for pJava specifications
Wireless Java for Symbian Devices	http://www.symbian.com/books/wjsd/wjsd-info.html
Java on Symbian	http://www.symbian.com/technology/standard-java.html

Links

Symbian C++ developer	http://www.symbian.com/developer/index.html (Be aware that this is not P800, P900 or P910 series specific information)
Sony Ericsson Developer World	http://www.sonyericsson.com/developer

Tips, information and external documentation

Development books

Author	Title	ISBN	Publication date
Allin	Wireless Java for Symbian Devices	0471486841	08/24/01
Dreamtech	Programming for embedded systems: Cracking the code	764549545	July 2002
Harrison	Symbian OS C++ for Mobile Phones	0470856114	04/17/03
Jipping	Symbian OS Communications Programming	0470844302	06/18/02
Mallick	Mobile And Wireless Design Essentials	0471214191	04/25/03
Symbian	Newcomer to Symbian OS Guide		

Index